

# Web Application for Large-Scale Multidimensional Data Visualization

Gintautas Dzemyda, Virginijus Marcinkevičius and  
Viktor Medvedev

*Institute of Mathematics and Informatics, Vilnius University*

Akademijos 4, LT-08663, Vilnius, Lithuania

E-mail: [Gintautas.Dzemyda@mii.vu.lt](mailto:Gintautas.Dzemyda@mii.vu.lt)

E-mail: [Virginijus.Marcinkevicius@mii.vu.lt](mailto:Virginijus.Marcinkevicius@mii.vu.lt)

E-mail(*corresp.*): [Viktor.Medvedev@mii.vu.lt](mailto:Viktor.Medvedev@mii.vu.lt)

Received September 1, 2010; revised March 22, 2011; published online May 1, 2011

**Abstract.** In this paper, we present an approach of the web application (as a service) for data mining oriented to the multidimensional data visualization. This paper focuses on visualization methods as a tool for the visual presentation of large-scale multidimensional data sets. The proposed implementation of such a web application obtains a multidimensional data set and as a result produces a visualization of this data set. It also supports different configuration parameters of the data mining methods used. Parallel computation has been used in the proposed implementation to run the algorithms simultaneously on different computers.

**Keywords:** Parallel computation, dimensionality reduction, computational methods, optimization, multidimensional data.

**AMS Subject Classification:** 65Y05; 90-08.

## 1 Introduction

Interaction between humans and machines is one of the areas in computer science that has evolved a lot in the last years. Progresses and innovations are mainly due to increases in computer power and technology of interactive software. Real data of natural and social sciences are often high-dimensional [2, 7, 9]. So, it is very difficult to understand these data and extract patterns. One way for such understanding is to make a visual insight into the analyzed data set [4, 8, 16]. To analyze multidimensional data we often use one of the main instruments of data analysis – data visualization or graphical presentation of information. The fundamental idea of visualization is to provide data in the form that would let the user to understand the data, to draw conclusions, and to influence directly a further process of decision making. Visualization allows a better comprehension of complicated data sets, it may help determine their subsets that interest the researcher. However, its results may be used in

decision making of various nature [2, 23]. Data visualization is closely related to dimensionality reduction methods that allow discarding interdependent data parameters, and by means of projection methods it is possible to transform multidimensional data to a line, plane, 3D space or other form that may be comprehended by a human eye. It is much quicker and easier to comprehend visual information than numerical or textual.

Data visualization allows people to detect the presence of clusters, outliers or regularities in the analyzed data. Let us have points

$$X^i = (x_1^i, x_2^i, \dots, x_n^i), \quad i = 1, \dots, m, \quad X^i \in R^n.$$

Denote the whole analyzed data set by  $X = \{X^1, \dots, X^m\}$ . The pending problem is to get the projection of points  $X^i \in R^n$ ,  $i = 1, \dots, m$  into points  $Y^1, Y^2, \dots, Y^m \in R^2$  corresponding to them. Here  $Y^i = (y_1^i, y_2^i)$ ,  $i = 1, \dots, m$ . In Fig. 1, we present an example of visual presentation of the data table (initial dimensionality  $n=7$ , the number of points  $m=70$ , and only the first 12 points are presented in Fig. 1) using the well-known Multidimensional scaling method [4]. The dimensionality of data is reduced from 7 to 2. Here some points form a separate cluster that can be clearly observed visually on a plane and that cannot be recognized directly from the table without a special analysis (see Fig. 1). Therefore, the goal of the projection methods is to represent the input data items in a lower-dimensional space so that certain properties of the structure of the data set were preserved as faithfully as possible [8].

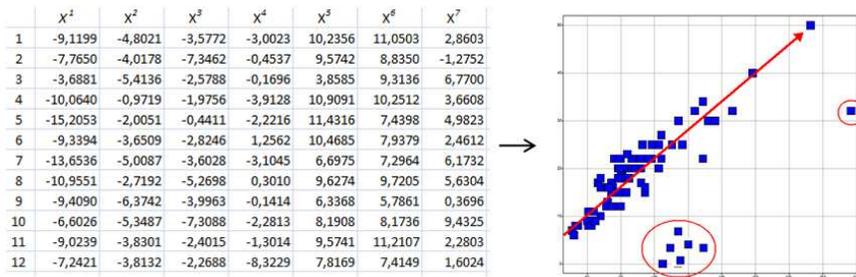


Figure 1. Visual presentation of the multidimensional data.

Visualization of large-scale multidimensional data can be combined with new ways of interacting with a computer using web service. Web services are typically the application of programming interfaces that are accessed via HyperText Transfer Protocol and executed on a remote system hosting the requested services [11]. Web services (or web applications) are self-contained, self-describing modular applications that can be published, located, and invoked across the web [3, 14]. Web services refer to a set of software applications or components, developed using a specific set of application programming interface standards and Internet-based communication protocols. The objective is to enable these applications or components to invoke function calls and exchange data among themselves over the standard Internet infrastructure. Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. The

web services architecture is an interoperability architecture: it identifies those global elements of the global network that are required in order to ensure interoperability between web services. By integrating dimensionality reduction algorithms, web interface, MPI (Message Passing Interface), a cluster for parallel computing into a multidimensional data visualization system, the users are able to identify the structure of the data and to optimize the visualization parameters in more simple way.

By integrating Internet technologies into multidimensional data visualization systems, we can get better performance results with additional functionalities. Providing seamless access to systems' functionality without downloading the software is the main concept behind web applications.

In this paper, we focus on the idea of the web application which provides multidimensional data visualization functionality. We try to combine the well-known visualization methods with modern technologies including the web-based service architectures and parallel computing.

## **2 Advantages of the Web Application**

Visualization is a useful way for data analysis, especially when the data come from the real world and, therefore, similarities of separate data items, characterized by a set of features (parameters), are unknown. The goal of the visualization methods is to represent the input data items in a lower-dimensional space so that certain properties of the structure of the data set were preserved as faithfully as possible. Nowadays, computer systems store large amounts of data. Visualization methods are computationally very intensive for large data sets. Due to the lack of abilities to adequately explore the large amounts of collected data, even potentially valuable data become useless. Therefore, we need the power of the parallel computing cluster to explore and analyze the collected data to solve the problem of large-scale multidimensional data visualization.

Several systems are proposed that provide web-based distributed visualization methods, e.g., Weka4ws (<http://grid.deis.unical.it/weka4as/>), Faehim (<http://users.cs.cf.ac.uk/Ali.Shaikhali/faehim>). These systems are rather powerful and include different data mining techniques. However, these systems require a specific knowledge and need to be installed in user's PC. The work with these systems is rather complicated, especially if we want to get a visualization of the data set analyzed without installing specific software and using only the Internet. In our approach, the access to the visualization service is possible from any location with internet connectivity independently of the used platform. The computational work is done using a high-performance parallel cluster, with only user's interaction with the client (without downloading and installing the system).

The web application for multidimensional data visualization provides a web-based access to several visual data mining methods of different nature and complexity that, in general, allows a visual discovery of patterns and their interpretation in multidimensional data. The developed software tool allows users to analyze and visualize large-scale multidimensional data sets on the Internet, regardless of time or location, as well as to optimize the parameters

of visualization algorithms for better perception of the multidimensional data.

To achieve a high performance in a large-scale multidimensional data visualization, the parallel computing has to be exploited. Parallel applications, including scientific applications, are now widely executed on clusters and grids [5]. For the large-scale multidimensional data visualization, a high-performance parallel cluster has been used in our implementation. The cluster differs from the network of workstations in security, application software, administration, and file systems. The important feature of the cluster is that it may be upgraded or expanded without any essential reconstructions.

The proposed web application simplifies the usage of three visualization methods and makes them wide-accessible: SMACOF Algorithm [4] - a version of Multidimensional Scaling (MDS), Relative MDS [20], and Diagonal Majorization Algorithm (DMA) [21].

### 3 Visualization Methods

We use nonlinear dimensionality reduction methods as a tool for visualizing large-scale multidimensional data sets. Several approaches have been proposed for reproducing nonlinear higher-dimensional structures on a lower-dimensional display. The most common methods determine a representation of each data point in a lower-dimensional space and try to optimize these representations so that the distances between them were as similar as possible to the original distances of the corresponding data items. The methods differ in that how the representations are optimized. *Multidimensional scaling* (MDS) [4] refers to a group of methods that is wide used. The starting position of MDS is a matrix consisting of pairwise dissimilarities of the entities. In general, the dissimilarities need not be distances in the mathematically strict sense. There exists a multitude of variants of MDS with slightly different cost functions and optimization algorithms.

The goal of projection in the metric MDS (when dissimilarities between objects are measured by some metric or distance function) is to optimize the projection so that the distances between the items in the lower-dimensional space would be as close to the original distances as possible. An objective function (stress) to be minimized can be written as

$$E_{MDS} = \sum_{i,j=1, i < j}^m w_{ij} (d_{ij}^* - d_{ij})^2. \quad (3.1)$$

Here  $d_{ij}^*$  is the distance between the points  $X^i$  and  $X^j$ , and  $d_{ij}$  is the distance between the corresponding points  $Y^i$  and  $Y^j$  in the projected space. Weights  $w_{ij}$  can be defined as follows:

$$w_{ij} = \frac{1}{\sum_{k,l=1, k < l}^m (d_{kl}^*)^2}, \quad w_{ij} = \frac{1}{d_{ij}^* \sum_{k,l=1, k < l}^m (d_{kl}^*)^2}, \quad w_{ij} = \frac{1}{m d_{ij}^*}.$$

Various types of minimization of the stress function are possible: a gradient descent, SMACOF (Scaling by MAjorization a COmplicated Function) [4],

Conjugate Gradient, Quasi-Newton Method, Simulated Annealing [15], Branch and Bound algorithm [1], Combination of the Genetic Algorithm and Quasi-Newton's Descent Algorithm [18].

The problems with the classical MDS algorithm are faced when we have to visualize a large data set or a new data point has to be projected among the previously mapped points. In the standard MDS, every iteration requires each point to be compared with all other points. Thus, the MDS method is unsuitable for large data sets: it requires very large CPU time or there is not enough computing memory. Various modifications of MDS have been proposed for visualization of large data sets: Steerable Multidimensional Scaling (MD-Steer) [22], Incremental MDS, Relative MDS [20], Landmark MDS [6], Diagonal Majorization Algorithm (DMA) [21], etc. In the web application proposed, the metric Multidimensional Scaling SMACOF algorithm has been used.

The SMACOF Algorithm (or Guttman Majorization Algorithm) is one of the best optimization algorithms for this type of minimization problem [10]. This method is simple and powerful, because it guarantees a monotone convergence of the stress function [4, 10]. It is possible to use the Guttman Majorization Algorithm, based on iterative majorization or its modification, i.e. the so-called *Diagonal Majorization Algorithm* (DMA).

Formula (3.2) is called the Guttman transform [19]:

$$Y(t' + 1) = V^\dagger B(Y(t')) Y(t'), \tag{3.2}$$

where  $B(Y(t'))$  has the elements:

$$b_{ij} = \begin{cases} -d_{ij}^*/d_{ij}, & \text{for } i \neq j \text{ and } d_{ij} \neq 0, \\ 0, & \text{for } i \neq j \text{ and } d_{ij} = 0, \\ b_{ii} = -\sum_{j=1, j \neq i}^m b_{ij}, & \text{for } i = j. \end{cases}$$

$V$  is the matrix of weights where  $V$  has the elements:

$$V = \begin{pmatrix} \sum w_{1j} & & & & \\ & \ddots & & -w_{ij} & \\ & & \ddots & & \\ & & & \ddots & \\ -w_{ij} & & & & \sum w_{mj} \end{pmatrix}, \quad i = 1, \dots, m.$$

$V^\dagger$  denotes the Moore-Penrose pseudoinverse of  $V$ . In our case,  $w_{ij}=1$ , therefore  $V^\dagger = m^{-1}I$ ; here  $I$  is an identity matrix.

The MDS algorithm minimizes the function (3.2) via the iterative process [4]. DMA was proposed in [21] and it uses the majorization function:

$$Y(t' + 1) = Y(t') + \frac{1}{2} \text{diag}(V)^{-1} [B(Y(t') - V)] Y(t') \tag{3.3}$$

with the weight matrix  $V$ , most values of which are equal to zero. Therefore, DMA attains a slightly worse projection error than the Guttman Majorization

Algorithm, but computing by the iteration equation (3.3) is faster and there is no need for computing the pseudoinverse matrix  $V^\dagger$  [21]. Iterative computations of coordinates  $Y^i = (y_1^i, y_2^i)$ ,  $i = 1, \dots, m$  are based not on all the distances  $d_{ij}^*$  between the multidimensional points  $X^i$  and  $X^j$ . This allows us to significantly speed up the visualization process and to save the computer memory essentially [17].

The MDS algorithm does not offer a possibility to project new points on the existing set of mapped points. In order to get a mapping that presents the previously mapped points together with the new ones, we should make a complete re-run of the MDS algorithm on the new and the old data points .

The main idea of the *Relative MDS* [20] method (which can be easily used for visualizing new points) is to take a subset of the initial multidimensional data set (basic data set) and then map the basic data set, using the MDS. As a second step, the remaining points of initial data are added to the basis layout using the relative mapping.

Let us denote the number of the known data points by  $m_F$ , the number of the new data points by  $m_M$ , the total number of points considered during the mapping by  $m$  ( $m = m_F + m_M$ ), the set of known data points by  $F$  (it will be called a set of basic points), the set of new data points by  $M$ . The algorithm scheme is as follows:

1. Map set  $F$  using the MDS mapping (the number of fixed points is equal to  $m_F$ ).
2. Map set  $M$  with respect to the mapped set  $F$ , using the relative mapping (the number of new points is equal to  $m_M$ ).

The difference between the relative mapping and the metric MDS is that during the minimization of the stress function, only the points from set  $M$  are allowed to move, while the points from set  $F$  are kept fixed. This is achieved by modifying the stress function so that it sums only over the distances that change during iterations, i.e., the distances between the fixed and the moving points, and interpoint distances between the moving points. The stress function (3.1) is rewritten as:

$$E_{Relative\_MDS} = \sum_{i,j=1, i < j}^{m_M} (d_{ij}^* - d_{ij})^2 + \sum_{i=1}^{m_F} \sum_{j=m_F+1}^{m_F+m_M} (d_{ij}^* - d_{ij})^2.$$

In the original Relative MDS algorithm, minimization of the projection error  $E_{Relative\_MDS}$  is also realized through the steepest descent procedure. However, it is not as effective as the Quasi-Newton algorithm. Therefore, in our web application, we use the Quasi-Newton algorithm to minimize  $E_{Relative\_MDS}$ .

The analyzed visualization algorithms are iterative or partially iterative which means that parallelization of these algorithms is not effective because of the expensive costs of data transmission between the processors of a parallel computing cluster in these algorithms. It was suggested to use a sequential version of algorithms and to apply MPI technology in order to make usage of multiple processors of the cluster possible. Each processor runs the algorithm from a different starting position.

## 4 Web Application for Multidimensional Data Visualization

The web service (web application) architecture [11] for multidimensional data visualization is a three-layer model (see Fig. 2). The Client Interface and Data Visualization Components layers are the main parts of the system. The client’s responsibility is to send data which must be accepted, processed and returned from the visualization service.

The Client Interface provides a way to define the processing routine for the given data in order to manage the visualization process. The client is responsible for the presentation of the data set to be visualized. Data Visualization Service is responsible to process data according to the created processing routine.

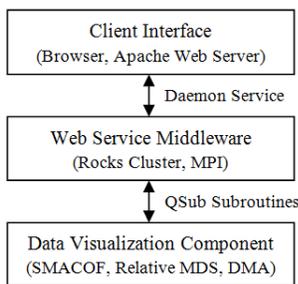


Figure 2. The web service architecture.

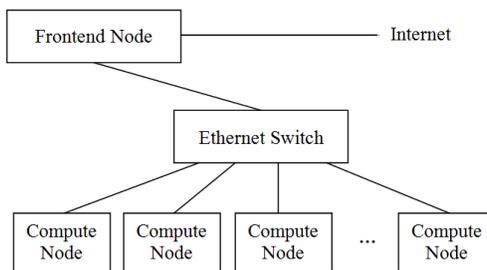


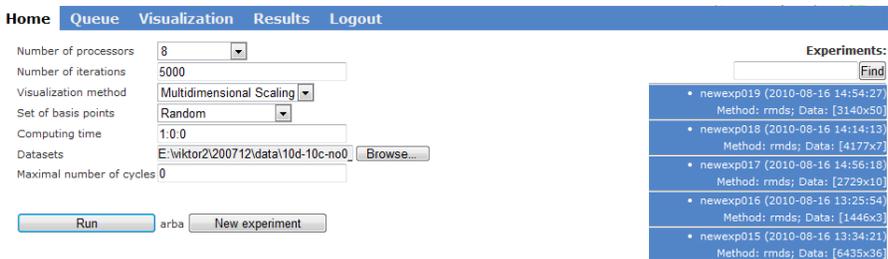
Figure 3. The Web Service Middleware structure.

The Data Visualization Component layer contains the algorithms meant for multidimensional data visual presentation. Since the visualization typically involves large-scale data sets, the efficiency of saving data can be extremely important. A computational cluster has been provided as the hardware system meant for performing visualization processes. In our case it is possible to run parallel visualization components that communicate through MPI (Message Passing Interface). MPI is a significant component of the programming and execution application on clusters. The analyzed visualization methods are based on iterative algorithms whose parallel versions are not effective. Using a cluster we can run the algorithms simultaneously on different computers from different random starting solutions. In this paper, we suggest to use the design and implementation of the Web Service Middleware that connects the Client Interface and the Data Visualization Component running on a computational cluster. The Web Service Middleware structure is presented in Fig. 3. It includes Frontend Node (Server) and the local computer network controlled by the Frontend Node. In our realization, the function of the local computer network is allocated to the computer cluster (for parallel computations). Such an architecture gives us an opportunity to solve large-scale data visualization problems, where a client does not care for computational resources and their proper usage.

At first, a client sends the data to the Data Visualization Component. In our case, three methods are included for visualizing multidimensional data: MDS (SMACOF algorithm), Relative MDS, and DMA. These methods have been chosen for testing the architecture. Relative MDS and DMA are designated to visualize large-scale multidimensional data. In future, the set of options for visualization may be extended.

In the Client Interface, it is possible to choose (and to change) the following parameters:

- Number of processors (in our implementation we can use 1–16 processors);
- Maximum number of iterations;
- Method for multidimensional data visualization (SMACOF algorithm, Relative MDS, DMA);
- Strategies of forming and initializing the set of basic points (on the line, random, maximal dispersion, principal component analysis);
- Maximal computing time (sometimes it is important to fix the computing time working with large data sets);
- Upload the client's data set for visualization (a text file containing a table of real numbers –  $m$  rows and  $n$  columns);
- Maximal number of visualization cycles (the current problem may be solved for several times with different initial data and the best result is presented to the client).



**Figure 4.** Client interface: visualization methods and settings.

Fig. 4 demonstrates some main components in the visualization section. In the Client section, the user can check the status of his task on the cluster.

Figures 5–6 show how the visualization process presents itself to the user. Two data sets were used to estimate the abilities of web application:

- Ellipsoidal data set [12], where  $m=3140$ ,  $n=50$ ; the set contains 10 overlapping ellipsoidal-type clusters. The data set have been visualized using Relative MDS using 10% of the basic points (Fig. 6a);

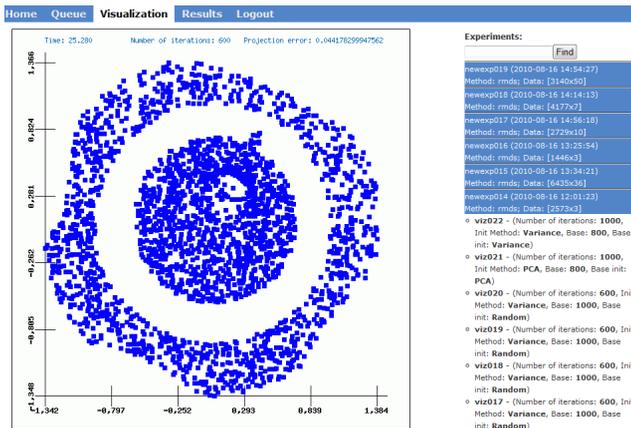


Figure 5. Client interface: visualization results.

- Sphere data set (three 3-dimensional spheres), where  $m=1446, n=3$  which contains three clusters. The data set has been visualized using Relative MDS with 10% of the basic points (Fig. 6b).

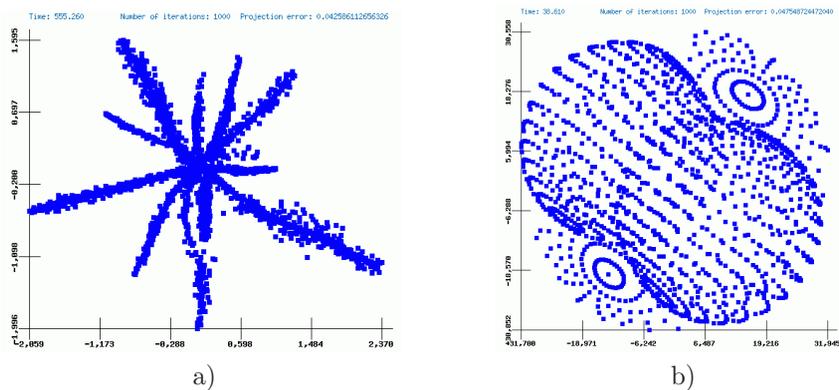
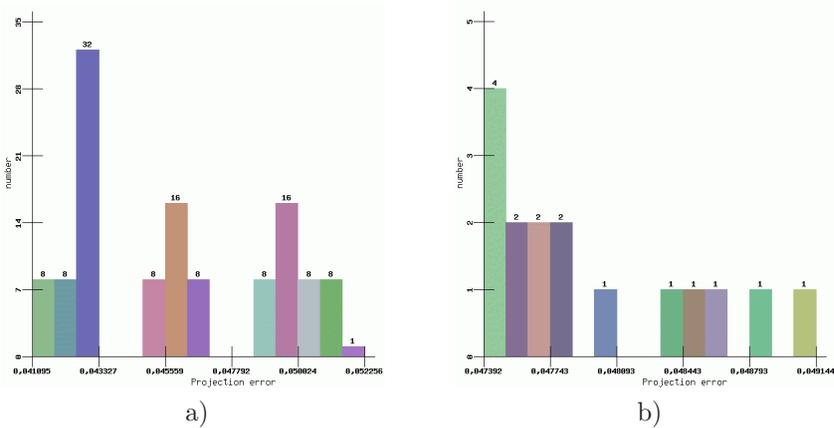


Figure 6. Visualization results (using Relative MDS): projections of the Ellipsoidal (a) and Sphere (b) data sets. The data sets were visualized with 10% of the analyzed data.

The user can get more information on the analyzed data set in the Results section:

- Distribution of the projection error on the basis of the fixed number of experiments (see Fig. 7). In the histograms (Fig. 7), we can see the data set projection error distribution and its changes with an increase in the number of iterations of the projection algorithm. In Fig. 7a, the local minimum of the projection error function with some fixed parameters of the visualization algorithm is presented. The parameters whose values may be set in the Client interface can be optimized in the case where we need to get quite a small local minimum. The distribution of the



**Figure 7.** Distribution of the projection error on the basis of the fixed experiments (using Relative MDS, PCA initialization method, 1000 iterations): (a) Ellipsoidal, (b) Sphere data sets.

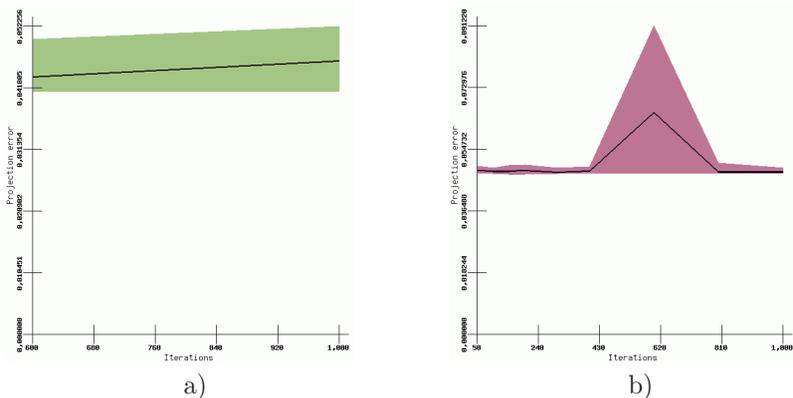
projection error indicates that the optimization by the Relative MDS reached at least three local minima. The reason is due to the complicated structure of the Ellipsoidal data set analyzed.

- Dependence of the projection error on the iteration number (see Fig. 8). In the exploration of projection error distribution, it is important to see the error distribution boundaries when the number of iterations increases. In Fig. 8a, the error boundaries with almost the same width and mean error (line) slightly move to the middle. In Fig. 8b, if the iteration number is equal to 600, then we can observe a distortion of projection error boundaries.
- Dependence of the computing time on the iteration number (see Fig. 9). For MDS and DMA algorithms, with an increase the iteration number, the computing time also linearly increases. Therefore, the processors of the cluster during experiments are loaded differently, or when a huge amount of memory is required, we get a dispersion of computing time (see Fig. 9a).

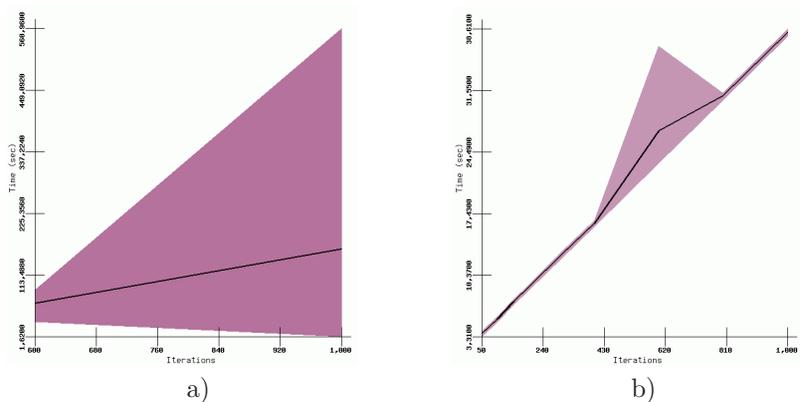
## Conclusions

In this paper, an approach and architecture have been proposed for visualization of large-scale multidimensional data, using the web service technology. This should extend the practical application of multidimensional data analysis and, particularly, visualization techniques.

The paper focuses on visualization methods as a tool for the visual presentation of large-scale multidimensional data sets. The web application proposed simplifies the usage of visualization methods that are often very sophisticated. In our case, three methods for visualizing multidimensional data are used:



**Figure 8.** Dependence of the projection error on the iteration number on the basis of the fixed experiments (using Relative MDS, PCA initialization method, 1000 iterations): (a) Ellipsoidal, (b) Sphere data sets.



**Figure 9.** Dependence of the computing time on the iteration number on a basis of the fixed experiments (using Relative MDS, PCA initialization method, 1000 iterations): (a) Ellipsoidal, (b) Sphere data sets.

MDS (SMACOF algorithm), Relative MDS, and DMA. These methods have been chosen for testing the architecture and approach. In future, the set of options for visualization should be extended. For example, recently new trends in multidimensional data visualization have become popular and found applications: various architectures of neural networks [8, 16] and nonlinear manifold learning methods [13].

The main advantage of the approach proposed is that it stimulates the visual data mining and pattern recognition in large-scale multidimensional data sets.

## Acknowledgement

This research was funded by a grant (No. TAP-43/2010) from the Research Council of Lithuania.

## References

- [1] M. Baravykaitė, R. Čiegis and J. Žilinskas. Template realization of generalized branch and bound algorithm. *Math. Model. Anal.*, **10**(3):217–236, 2005. Doi:10.1080/13926292.2005.9637283.
- [2] J. Bernatavičienė, G. Dzemyda, O. Kurasova, V. Marcinkevičius and V. Medvedev. The problem of visual analysis of multidimensional medical data. *Optimization and Its Applications*, **4**:277–298, 2007. Doi:10.1007/978-0-387-36721-7\_17.
- [3] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris and D. Orchard. Web service architecture. *W3C Working Group Note*, 2004.
- [4] I. Borg and J.P. Groenen. *Modern Multidimensional Scaling: Theory and Applications, second ed.* Springer, New York, 2005.
- [5] R. Čiegis, A. Jakušev, A. Krylovas and O. Suboč. Parallel algorithms for solution of nonlinear diffusion problems in image smoothing. *Math. Model. Anal.*, **10**(2):155–172, 2005. Doi:10.1080/13926292.2005.9637279.
- [6] V. de Silva and J.B. Tenenbaum. Global versus local methods for nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, **15**:721–728, 2003.
- [7] G. Dzemyda. Visualization of a set of parameters characterized by their correlation matrix. *Computational Statistics and Data Analysis*, **36**(10):15–30, 2001. Doi:10.1016/S0167-9473(00)00030-X.
- [8] G. Dzemyda, O. Kurasova and V. Medvedev. Dimension reduction and data visualization using neural networks. In *Emerging Artificial Intelligence Applications in Computer Engineering – Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, volume 160 of *Frontiers Artificial Intelligence Appl.*, pp. 25–49, 2007.
- [9] A. Frank and A. Asuncion. *UCI Machine Learning Repository*. University of California, School of Information and Computer Science, Irvine, CA, 2010. Available from Internet: <http://archive.ics.uci.edu/ml>.
- [10] P.J.F. Groenen and M. van de Vaelden. Multidimensional scaling. *Econometric Institute Report EI2004-15*, 2004.
- [11] J. Hagel. Edging into Web services. *Quarterly*, **4**:29–38, 2002.
- [12] J. Handl and J. Knowles. *Cluster generators for large high-dimensional data sets with large numbers of clusters*, 2005. Available from Internet: <http://dbkgroup.org/handl/generators/>.
- [13] R. Karbauskaitė and G. Dzemyda. Topology preservation measures in the visualization of manifold-type multidimensional data. *Informatika*, **20**(2):235–254, 2009.
- [14] K.M. Khan. *Managing Web Service Quality: Measuring Outcomes and Effectiveness, Information Science Reference*. Qatar University, Qatar, 2008.
- [15] H. Klock and J. M. Buhmann. Data visualization by multidimensional scaling: A deterministic annealing approach. *Pattern Recognition*, **33**(4):651–669, 1999. Doi:10.1016/S0031-3203(99)00078-3.
- [16] T. Kohonen. *Self-Organizing Maps, third ed.* Springer, 2001.

- [17] V. Marcinkevičius. *Investigation and Functionality Improvement of Nonlinear Multidimensional Data Projection Methods*. Kauno technologijos universiteto spaustuvė, Vilnius, 2010. (in Lithuanian, PhD thesis)
- [18] R. Mathar and A. Žilinskas. On global optimization in two-dimensional scaling. *Acta Appl. Math.*, **33**:57–64, 2001.
- [19] A. Morrison, G. Ross and M. Chalmers. Fast multidimensional scaling through sampling, springs and interpolation. *Information Visualization*, **2**(1):68–77, 2003. Doi:10.1057/palgrave.ivs.9500040.
- [20] A. Naud and W. Duch. Interactive data exploration using MDS mapping. In *Proceedings of the Fifth Conference: Neural Networks and Soft Computing*, pp. 255–260, 2000.
- [21] W.M. Trosset and P.J. Groenen. Multidimensional scaling algorithms for large data sets interactive data exploration using MDS mapping. In *Proceedings of the Computing Science and Statistics*, 2005.
- [22] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *Information Visualization, 2004*, INFOVIS 2004. IEEE Symposium, pp. 57–64, 2004.
- [23] E.K. Zavadskas, A. Kaklauskas, Z. Turskis and J. Tamošaitienė. Multi-attribute decision-making model by applying grey numbers. *Informatika*, **20**(2):305–320, 2009.